



SCHOOL OF SCIENCE & ENGINEERING SPRING 2023

CSC 3326 Database Systems

Term Project: Final Report

March 22nd, 2023

Realized by:

Hiba Msatfa

Adnane Ahroum

Chaima Aissi

Supervised by:

Professor BOUANANE, Lamiae

Table of content:

I. Cover page

II. Table of content

III. Introduction

A. Project description: Organizational structure, mission, operations, problems and constraints

B. Project objectives

C. Project Planning

IV. Design

A. Conceptual design

1. Requirements' specification: main system processes

2. Business rules, entities, relationships

3. Initial E-R model

4. Table normalization

5. Model verification: data manipulation operations and queries

6. Final E-R model

B. Logical design: tables, attributes' data types, constraints and views

V. Implementation

A. Database tables creation and population

B. Application architecture: technology and tools used and their interaction/web user-friendly interface

VI. Testing and fine-tuning

VII. User manual: Description of menus, forms, screens' snapshots...

VIII. Conclusion

IX. Future work (proposed future work to extend/improve your application)

I. Introduction:

A. Project description: Organizational structure, mission, operations, problems and constraints:

GoldenBlood is a blood center involved in blood donation and sales. To overcome existing challenges, our team plans to develop a web application for GoldenBlood. This application aims to enhance the blood donation process by increasing its security, reliability, and efficiency. The objectives include minimizing errors, reducing processing time, securing data, and improving information storage and retrieval for better overall efficiency.

The organizational structure that we followed is that we first use a password and an ID given by the agency to each of its clients to log in. The admin, which is the user, will be allowed to access the recipients, the donors, the inventory and the staff members. From then on, the user can make any changes needed, as well as add recipients; donors; products and even new employees to the system. The admin can also create appointments with the recipients and the donors.

Our main challenge was time. Throughout the semester, we struggled to find suitable meeting times to initiate our project due to heavy workloads from other classes. This delay caused us to start the project late, resulting in insufficient implementation of all our requirements. Additionally, during the implementation phase, we encountered difficulties connecting the database to the backend, leading to various issues. As a solution, we decided to explore alternative tools that would simplify this process, which will be elaborated on later. Lastly, we faced difficulties with linking the front and back end, similar to the database problem, but fortunately, we discovered a tool that facilitated this task.

B. Project objectives:

The aim of this project is to manage data related to blood donors, recipients, and the current stock. The process involves an individual physically coming to the organization, either to donate blood or to request it. They will receive a unique identification number, a password, and a status, and after logging in to the web-based application, they can input all the required details. The administrators can then arrange an appointment with the individual.

We have identified the following objectives for the project:

- Create a dedicated webpage for Admins that grants them access to the inventory and allows them to schedule appointments with clients and donors as well as add any new Staff members.
- The website should maintain a record of the inventory, including the different blood types and the corresponding quantities available.

To summarize, the objectives of the product are as follows:

- Be more time efficient.
- Simplify the blood donation sign-up process and enable individuals to locate blood donors more easily.
- Streamline the process of connecting clients with blood donors by making the data more accessible and easier to retrieve.

C. Project Planning:

The operations of the project are as follows:

- **Login:**

Once the user (which are the employees and staff members) receives their login information from the company and logs into the website, they will need to provide two attributes: their ID and Password. The page will then load into an admin view.

- **Admin View:**

The admin view offers four options. Whether the staff member would like to check the inventory, the information about staff, the information about Recipients and the information about the Donors.

- **Recipient View:**

The page Recipient will offer two possibilities, whether to add a recipient or whether to create an appointment for a recipient.

- **Admin View:**

The page Admin will offer two possibilities, whether to add an admin or whether to create an appointment for an admin.

- **Staff View:**

The page Staff will allow the users to add new employees into the system.

- **Inventory View:**

The inventory view allows us to add new products into the inventory, as we specify and fill all of the necessary fields.

II. Design:

A. Conceptual design:

1. Requirements' specification: main system processes

In our project, we have established an organized and secure organizational structure that ensures effective management of the system. To access the system, each employee is provided with a unique password and ID assigned by the agency. By using this login information, employees can securely log into the system and gain access to their personalized accounts.

The primary user with administrative privileges, known as the admin, plays a crucial role in overseeing and managing the system. The admin has comprehensive access to different modules within the system, including recipients, donors, inventory, and staff members. This level of access allows the admin to view, modify, and update information related to recipients, donors, inventory items, and employees as per the organization's requirements.

The admin possesses the capability to make any necessary changes to the system. This includes adding new recipients to the database, registering new donors, managing inventory items such as blood products, and even adding new employees to the system. By empowering the admin with these capabilities, the system becomes highly adaptable and can accommodate the organization's evolving needs.

Additionally, the admin has the authority to create appointments with both recipients and donors. This functionality streamlines the process of scheduling and coordinating blood donations or distributions. By facilitating appointment creation, the admin ensures efficient utilization of resources and enables better planning and coordination within the organization.

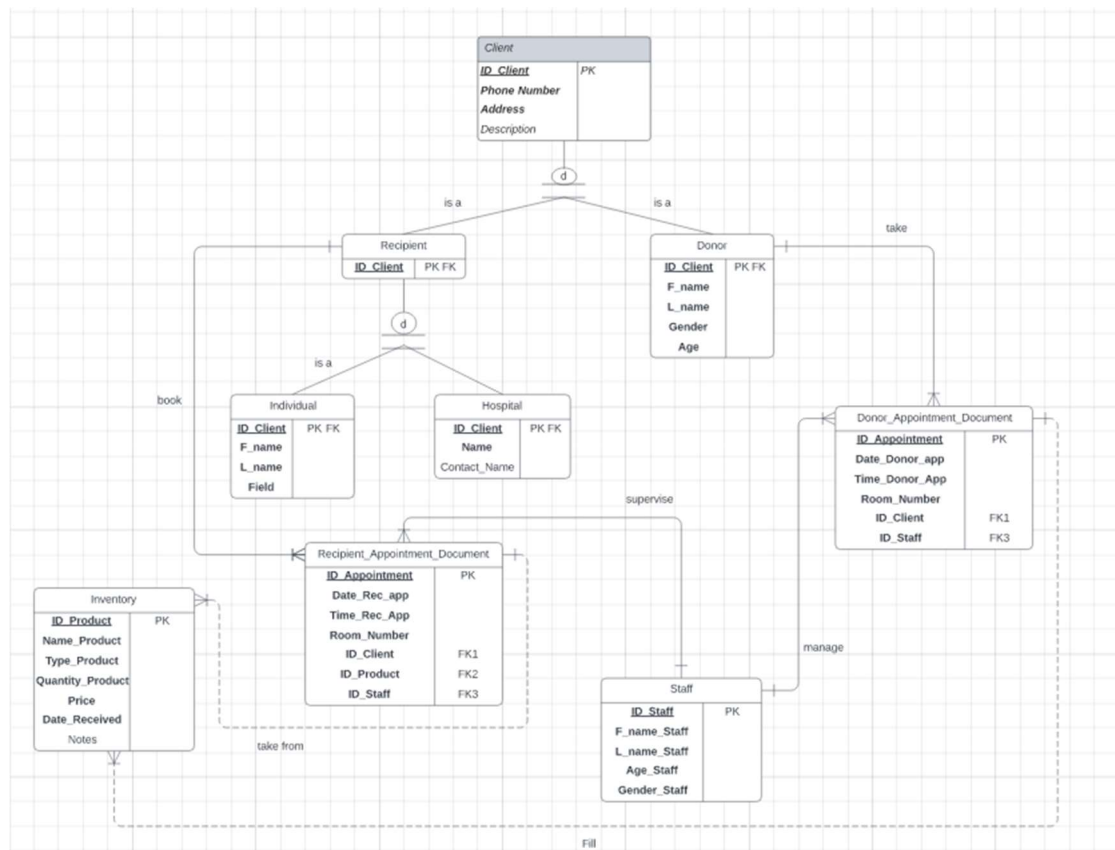
2. Business rules, entities, relationships

Our project entities are as follows: Client, Recipient, Donor, Staff, Inventory, Recipient Appointment, Donor Appointment.

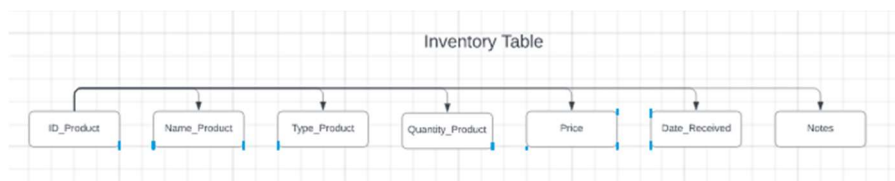
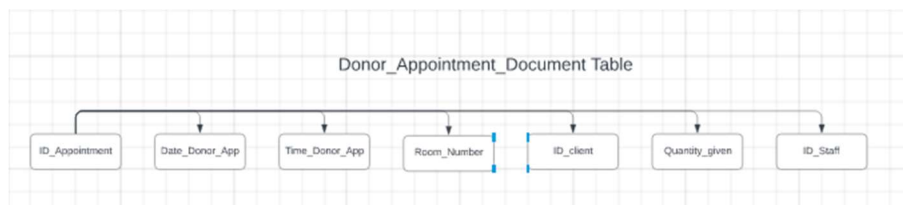
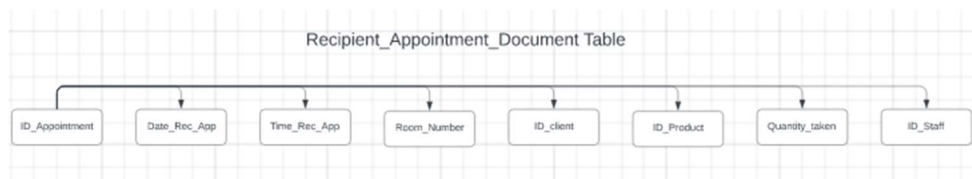
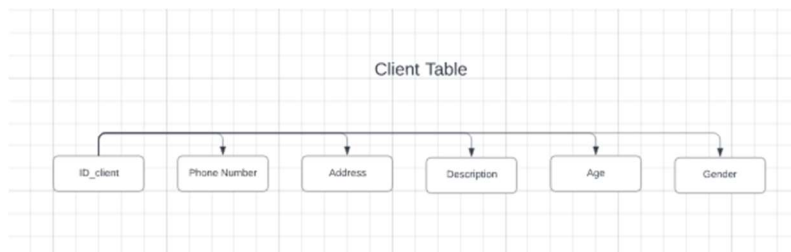
The business rules are as follows:

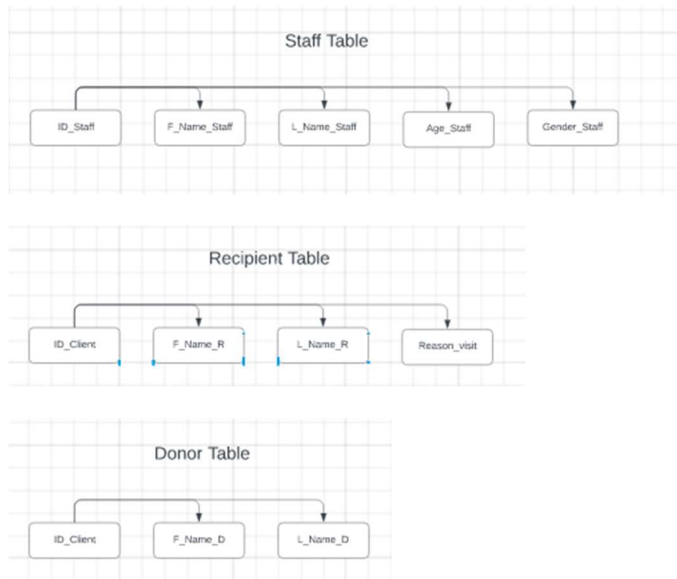
- A user can either be a recipient or a donor.
- An individual can come and ask for blood for personal issues.
- A client has an ID, a phone number, an address, a name, age and gender as well as a description where we can add any relevant details about the client, but this last attribute can be optional.
- Our inventory contains the products, their names, their specific IDs, the quantity on hand of said product as well as the date it was received and its price. We can also add some notes if they were relevant.
- We also have our staff which have their names as well as age and gender.
- For the recipient to receive blood they need an appointment.
- For the donor to give blood we also need an appointment.
- A recipient can make many appointments, but an appointment can only be booked by one person at a time.
- Thanks to the donor appointments, we can fill our inventory after further inspection of the product received.
- The recipient appointments take directly from the inventory.
- A donor can make many appointments but an appointment can only be made from one donor.
- One staff can supervise many appointments, but one appointment can only be supervised by one staff member.

3. Initial E-R model:



4. Table Normalization:





5. Model verification: data manipulation operations and queries

Data manipulation operations and queries are essential components of any data management system. These operations involve modifying, inserting, deleting, or retrieving data from a database. In our project, we have implemented a range of data manipulation operations and queries to enable efficient management and retrieval of information.

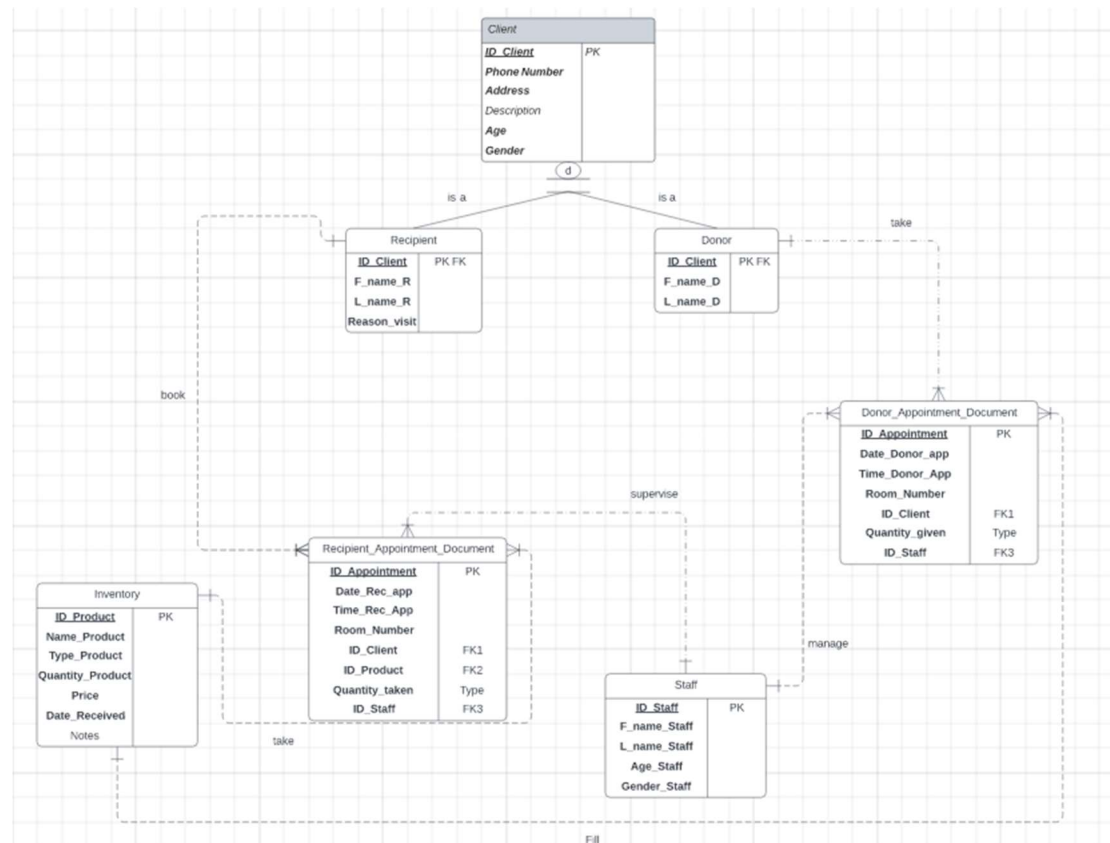
One of the key data manipulation operations we have incorporated is the ability to insert new data into the system. This allows administrators to add new recipients, donors, inventory items, and employees to the database. By providing an intuitive user interface, our system ensures that the required information can be easily entered and stored for future reference.

To maintain data integrity, we have also included delete operations, which allow administrators to remove outdated or unnecessary records from the system. This ensures that the database remains streamlined and avoids cluttering with redundant information.

In addition to data manipulation operations, our system supports various queries to retrieve specific information from the database. Administrators can execute queries to obtain reports on inventory levels, donor eligibility, recipient appointments, or any other relevant data. These queries provide valuable insights and aid decision-making within the organization.

Overall, our project incorporates robust data manipulation operations and queries, enabling administrators to efficiently manage and retrieve information from the system. These functionalities ensure data accuracy, integrity, and accessibility, empowering the organization to make informed decisions and streamline its blood management processes.

6. Final E-R model



B. Logical design: tables, attributes' data types, constraints and views

In our project, we have developed a logical design that encompasses tables, attributes' data types, constraints, and views. This design aims to efficiently organize and manage the data within the system, ensuring data integrity and facilitating data retrieval.

We have identified several tables to store different types of information. These tables include the recipient table, donor table, inventory table, staff table, donor appointment table and recipient appointment table. Each table represents a specific entity within the system and contains relevant attributes to capture the required data.

To enforce data integrity, we have defined constraints on the tables. These constraints ensure that the data entered into the system meets specific criteria. As an example, the gender in the Client table and in the Staff table can only take two values (F or M), or that the maximum quantity that we can hold of each product in the inventory is 5000.

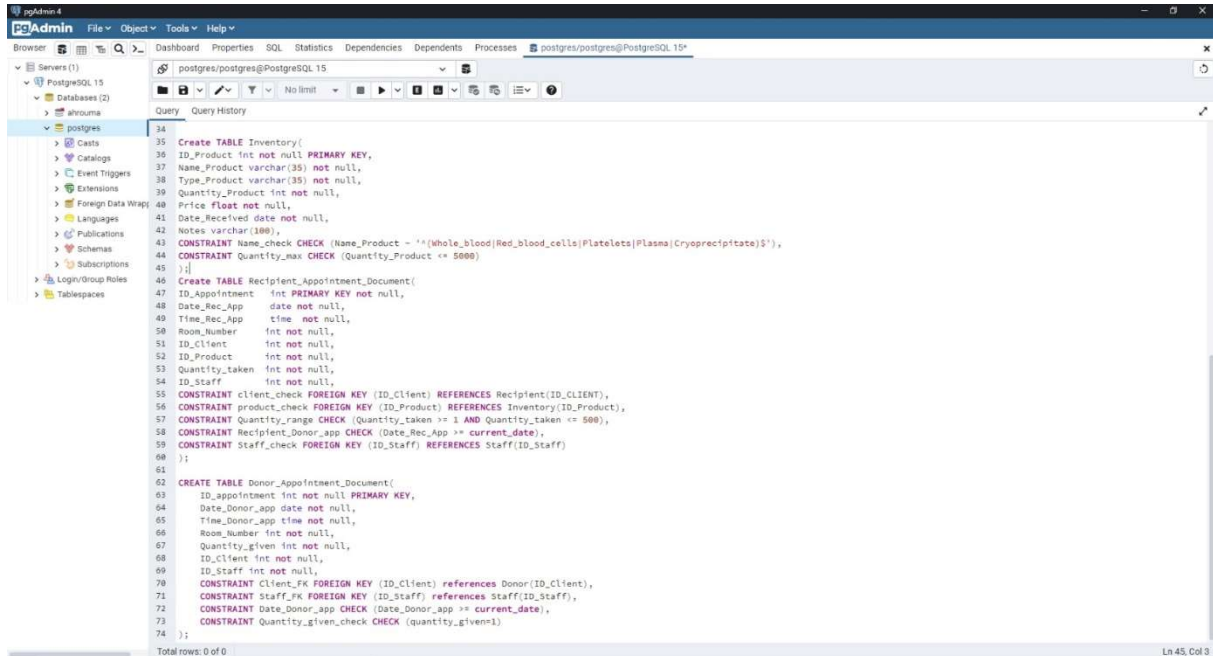
Views provide a customized and simplified perspective of the underlying data. In our system, we have created views to present consolidated information that meets specific requirements. For example, we may create a view that combines recipient and donor data for easier analysis and reporting. By utilizing views, users can access and manipulate the data in a more intuitive and meaningful manner.

Overall, our logical design encompasses well-defined tables, appropriate attribute data types, constraints to maintain data integrity, and views to provide customized data presentations.

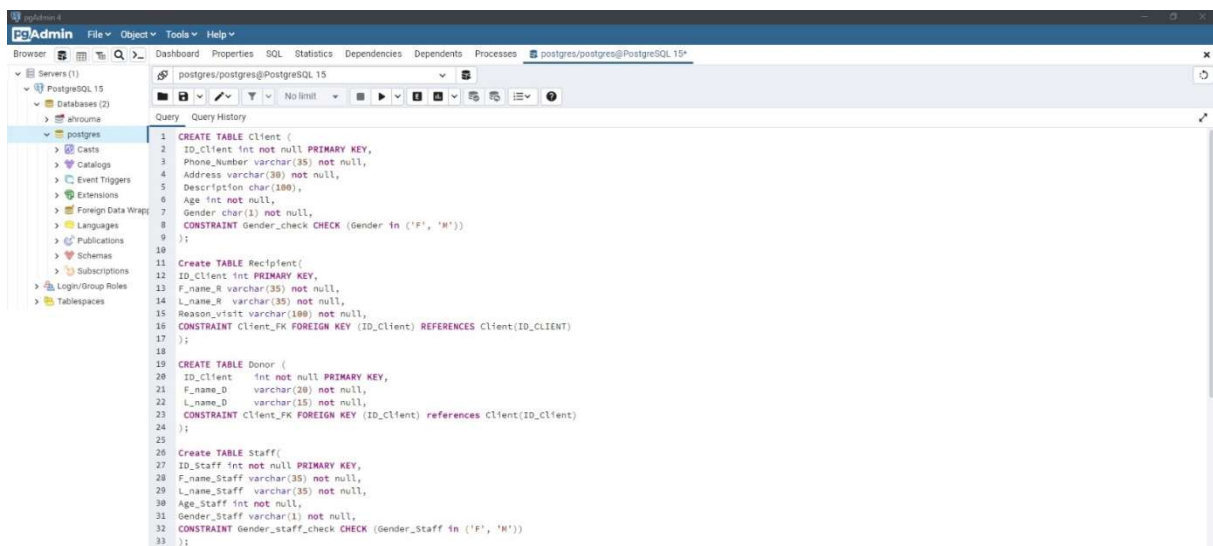
This design ensures efficient data organization, accurate storage, and retrieval, ultimately supporting effective management and analysis within the system.

III. Implementation:

A. Database tables creation and population:



```
34
35 Create TABLE Inventory(
36 ID_Product int not null PRIMARY KEY,
37 Name_Product varchar(35) not null,
38 Type_Product varchar(35) not null,
39 Quantity_Product int not null,
40 Price float not null,
41 Date_Received date not null,
42 Notes varchar(100),
43 CONSTRAINT Name_check CHECK (Name_Product ~ '^((Whole_blood|Red_blood_cells|Platelets|Plasma|Cryoprecipitate)$)'),
44 CONSTRAINT Quantity_max CHECK (Quantity_Product <= 5000)
45 );
46 Create TABLE Recipient_Appointment_Document(
47 ID_Appointment int not null,
48 Date_Rec_App date not null,
49 Time_Rec_App time not null,
50 Room_Number int not null,
51 ID_Client int not null,
52 ID_Product int not null,
53 Quantity_taken int not null,
54 ID_Staff int not null,
55 CONSTRAINT client_check FOREIGN KEY (ID_Client) REFERENCES Recipient(ID_Client),
56 CONSTRAINT product_check FOREIGN KEY (ID_Product) REFERENCES Inventory(ID_Product),
57 CONSTRAINT Quantity_range CHECK (Quantity_taken >= 1 AND Quantity_taken <= 500),
58 CONSTRAINT Recipient_Donor_app CHECK (Date_Rec_App >= current_date),
59 CONSTRAINT Staff_check FOREIGN KEY (ID_Staff) REFERENCES Staff(ID_Staff)
60 );
61
62 Create TABLE Donor_Appointment_Document(
63 ID_appointment int not null PRIMARY KEY,
64 Date_Donor_app date not null,
65 Time_Donor_app time not null,
66 Room_Number int not null,
67 Quantity_given int not null,
68 ID_Client int not null,
69 ID_Staff int not null,
70 CONSTRAINT client_fk FOREIGN KEY (ID_Client) references Donor(ID_Client),
71 CONSTRAINT Staff_fk FOREIGN KEY (ID_Staff) references Staff(ID_Staff),
72 CONSTRAINT Date_Donor_app CHECK (Date_Donor_app >= current_date),
73 CONSTRAINT Quantity_given_check CHECK (quantity_given > 0)
74 );
```



```
1 CREATE TABLE Client (
2 ID_Client int not null PRIMARY KEY,
3 Phone_Number varchar(35) not null,
4 Address varchar(30) not null,
5 Description char(100),
6 Age int not null,
7 Gender char(1) not null,
8 CONSTRAINT Gender_check CHECK (Gender in ('F', 'M'))
9 );
10
11 Create TABLE Recipient(
12 ID_Client int PRIMARY KEY,
13 F_name_R varchar(35) not null,
14 L_name_R varchar(35) not null,
15 Reason_Visit varchar(100) not null,
16 CONSTRAINT Client_FK FOREIGN KEY (ID_Client) REFERENCES Client(ID_Client)
17 );
18
19 Create TABLE Donor (
20 ID_Client int not null PRIMARY KEY,
21 F_name_D varchar(20) not null,
22 L_name_D varchar(15) not null,
23 CONSTRAINT client_fk FOREIGN KEY (ID_Client) references Client(ID_Client)
24 );
25
26 Create TABLE Staff(
27 ID_Staff int not null PRIMARY KEY,
28 F_name_Staff varchar(35) not null,
29 L_name_Staff varchar(35) not null,
30 Age_Staff int not null,
31 Gender_Staff varchar(1) not null,
32 CONSTRAINT Gender_staff_check CHECK (Gender_Staff in ('F', 'M'))
33 );
```

```
1 INSERT INTO Staff (ID_Staff, F_name_Staff, L_name_Staff, Age_Staff, Gender_Staff) VALUES
2 (1, 'John', 'Doe', 30, 'M'),
3 (2, 'Jane', 'Smith', 28, 'F'),
4 (3, 'Michael', 'Johnson', 35, 'M'),
5 (4, 'Emily', 'Davis', 32, 'F');
```

```

1 CREATE OR REPLACE FUNCTION update_inventory_quantity()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     IF TG_OP = 'INSERT' THEN
5         -- Decrease the quantity of the product in the inventory when a recipient appointment is scheduled
6         UPDATE Inventory
7         SET Quantity_Product = Quantity_Product - NEW.Quantity_taken
8         WHERE ID_Product = NEW.ID_Product;
9     ELSIF TG_OP = 'DELETE' THEN
10        -- Increase the quantity of the product in the inventory when a recipient appointment is canceled
11        UPDATE Inventory
12        SET Quantity_Product = Quantity_Product + OLD.Quantity_taken
13        WHERE ID_Product = OLD.ID_Product;
14    END IF;
15    RETURN NEW;
16 END;
17 $$ LANGUAGE plpgsql;
18
19 CREATE TRIGGER update_inventory_trigger
20 AFTER INSERT OR DELETE ON Recipient_Appointment_Document
21 FOR EACH ROW
22 EXECUTE FUNCTION update_inventory_quantity();
23

```

```

1 CREATE OR REPLACE FUNCTION update_donor_status()
2 RETURNS TRIGGER AS $$
3 BEGIN
4     IF TG_OP = 'INSERT' THEN
5         -- Update the donor status to 'Inactive' if their age is above 65
6         IF NEW.Age >= 65 THEN
7             UPDATE Donor
8             SET Status = 'Inactive'
9             WHERE ID_Client = NEW.ID_Client;
10        END IF;
11    END IF;
12    RETURN NEW;
13 END;
14 $$ LANGUAGE plpgsql;
15
16 CREATE TRIGGER update_donor_status_trigger
17 AFTER INSERT ON Donor
18 FOR EACH ROW
19 EXECUTE FUNCTION update_donor_status();
20

```

The only table that we need to populate is the Staff table, as the other tables will be filled as we use the web application. I populated the table with some example values.

B. Application architecture: technology and tools used and their interaction/web user-friendly interface:

For the database, we used elephant SQL to create our queries and populate them. ElephantSQL is a highly beneficial tool for creating databases due to its numerous advantages. Firstly, it offers a user-friendly interface that simplifies the process of setting up and managing databases, making it accessible even for those without extensive technical expertise. Additionally, ElephantSQL provides a cloud-based infrastructure, eliminating the need for users to set up and maintain their own database servers, reducing operational complexities and costs.

For the front end and back end, I used Goodot. As I stated earlier, we ran into an issue of time, therefore I needed to use a tool that will make the development easier for me in order to be able to submit the project in time. Goodot was a great help for this situation.

One of the main advantages of using Godot for web application development is its intuitive visual editor, which allows developers to create user interfaces and design interactive elements with ease. The engine supports multiple programming languages, including GDScript, C#, and VisualScript, providing flexibility for developers to work with their preferred language. Additionally, Godot offers extensive documentation and an active community, ensuring that developers can easily find resources and assistance when building web applications. The engine's lightweight nature and efficient performance make it well-suited for web-based projects, ensuring smooth user experiences. Moreover, Godot supports various web technologies, such as HTML5 and WebSockets, enabling developers to integrate web-specific features seamlessly.

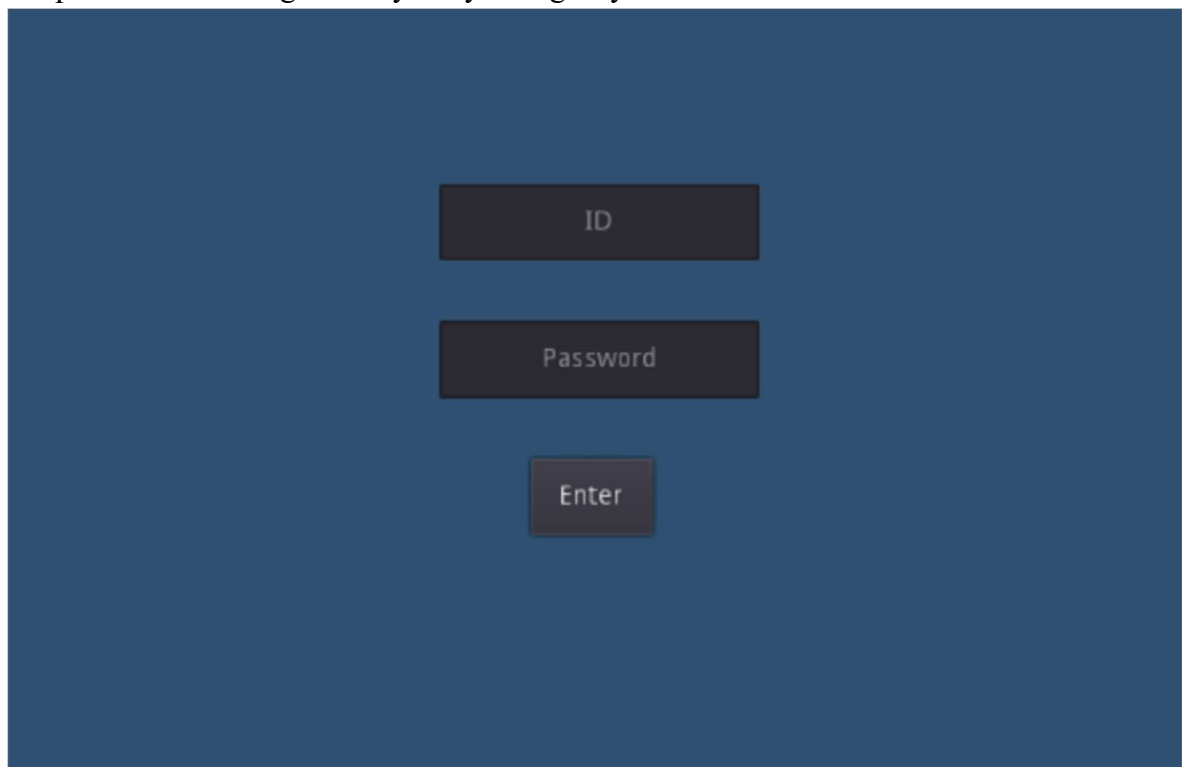
IV. Testing and fine-tuning:

Please watch the video demo for the testing of the web app.

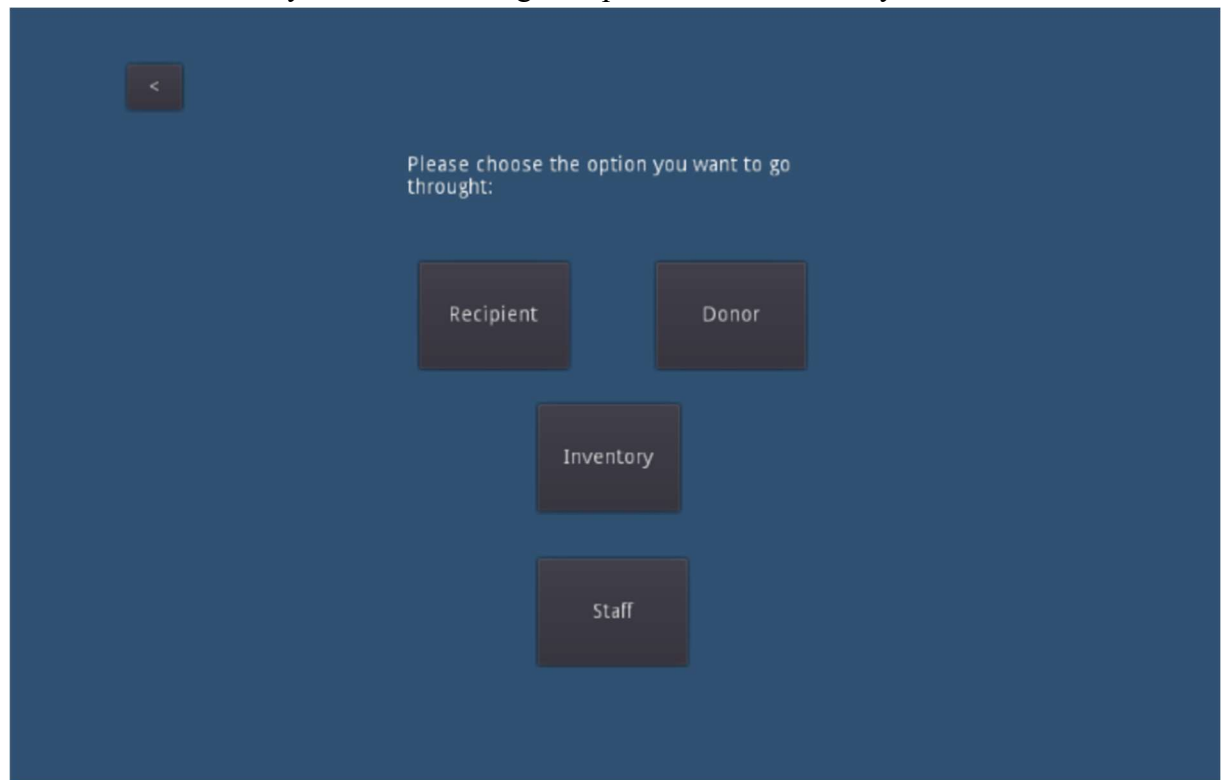
V. User manual: Description of menus, forms, screens' snapshots:

This user manual will guide you through the features and functionalities of our application. Please read this manual carefully to make the most out of your experience.

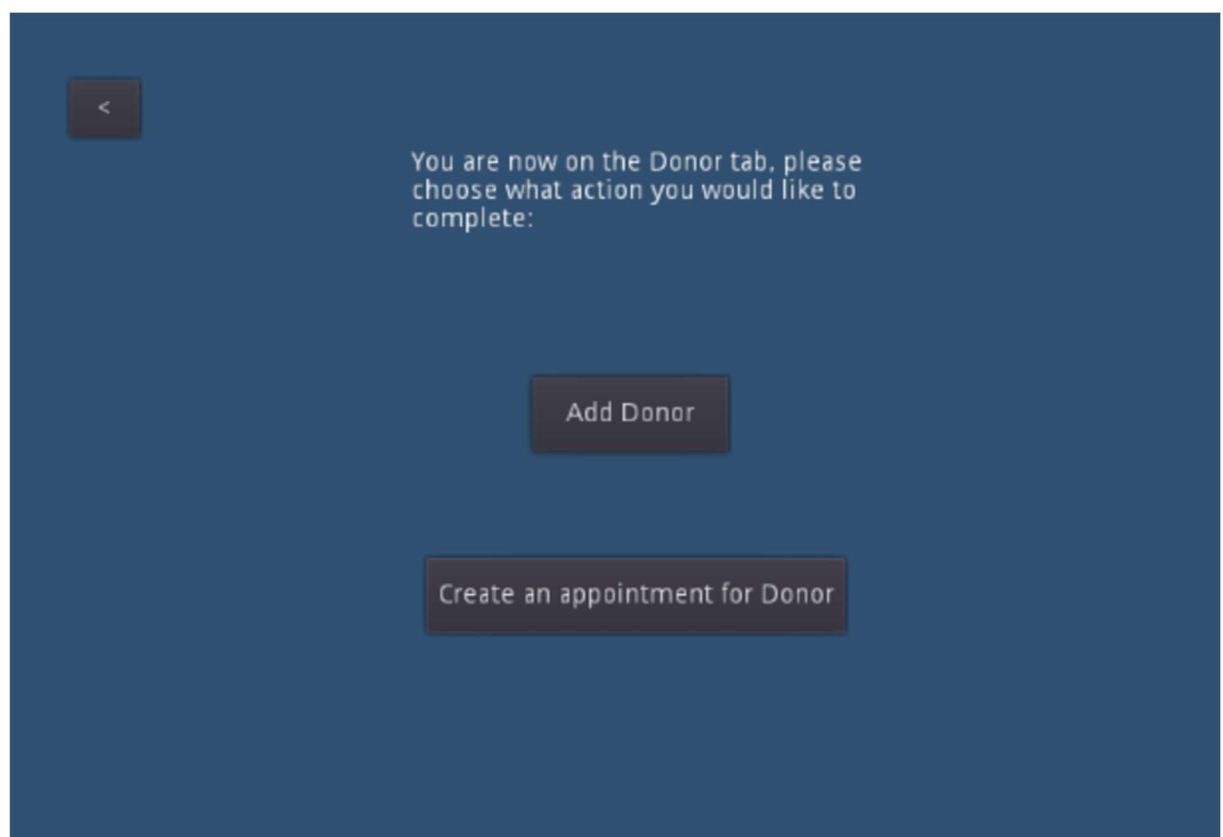
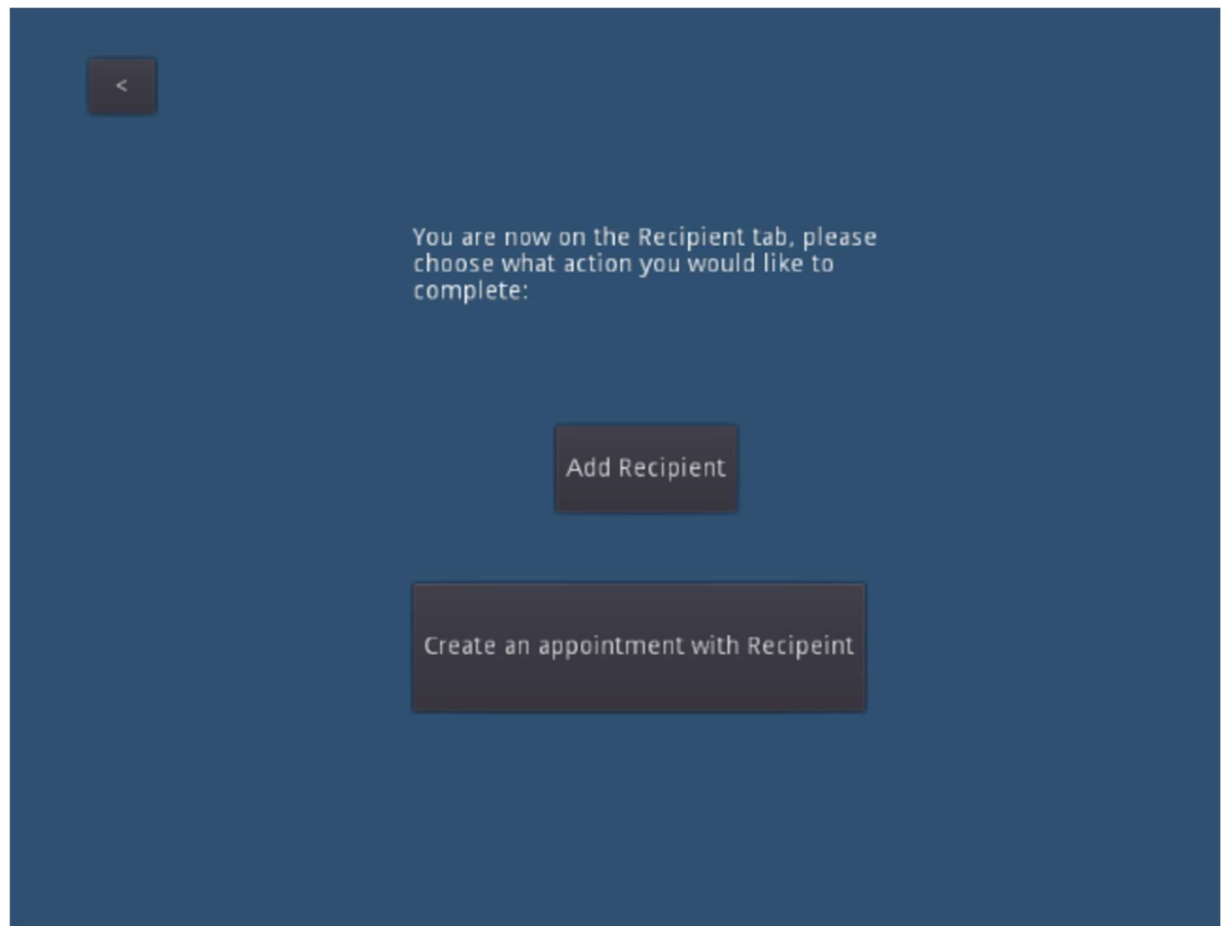
- The first thing you will see when you open the app is a log in page. Please enter the password and ID given to you by the agency.



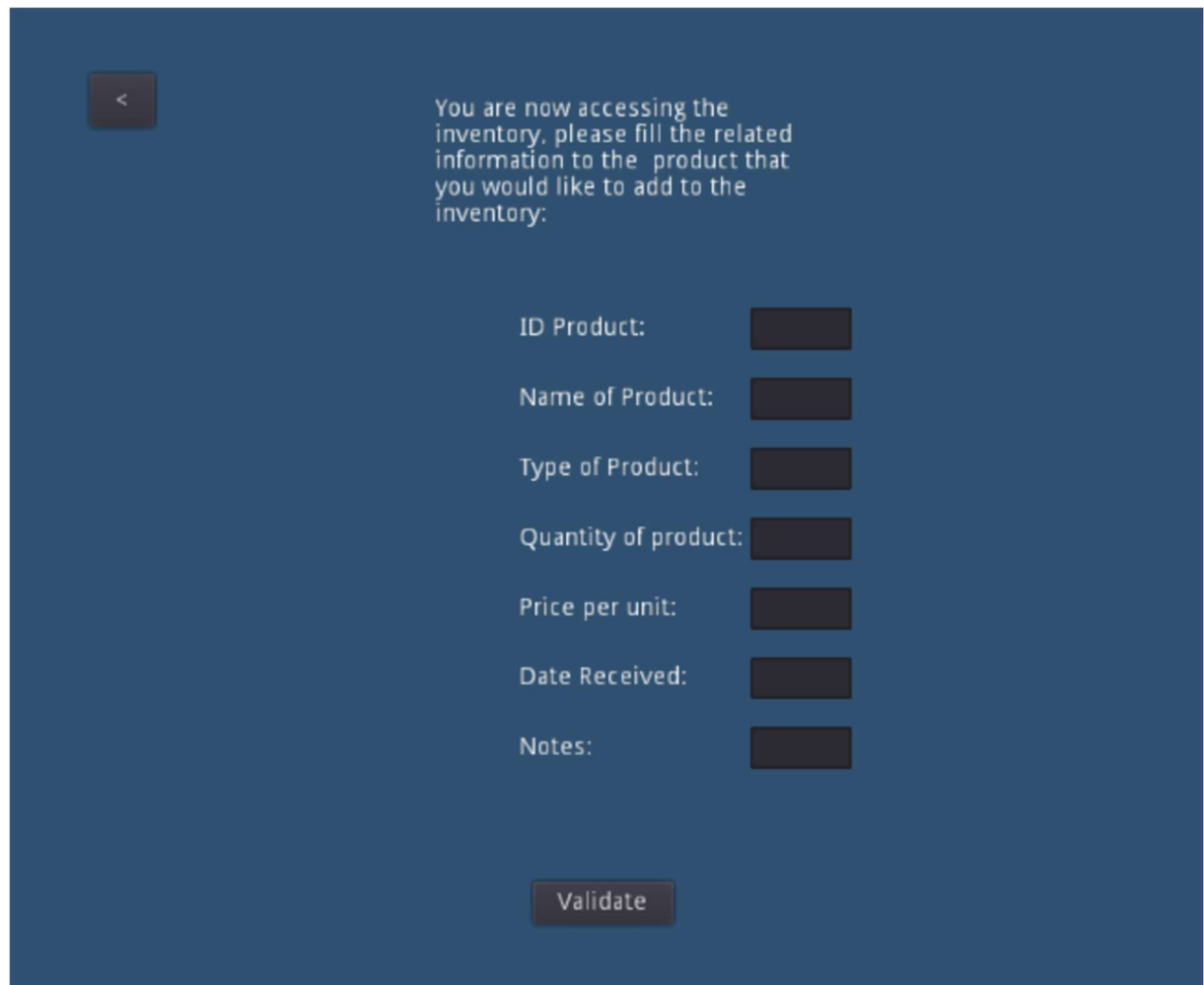
- Once you click on validate, a new page will load. The new page will give you 4 choices. Whether you want to manage recipient, donor, inventory or staff members.



- For recipient and donor, you can either add new ones to the database or create new appointments with said donors and recipients.



- For the inventory, you can add new products to the database.



The screenshot shows a web interface with a dark blue background. In the top left corner, there is a small dark button with a white left-pointing arrow. To its right, white text reads: "You are now accessing the inventory. please fill the related information to the product that you would like to add to the inventory:". Below this text, there are seven input fields, each preceded by a label in white text. The labels and their corresponding input fields are: "ID Product:" followed by a short text box; "Name of Product:" followed by a text box; "Type of Product:" followed by a text box; "Quantity of product:" followed by a text box; "Price per unit:" followed by a text box; "Date Received:" followed by a text box; and "Notes:" followed by a text box. At the bottom center of the form, there is a dark button with the word "Validate" in white text.

<

You are now accessing the inventory. please fill the related information to the product that you would like to add to the inventory:

ID Product:

Name of Product:

Type of Product:

Quantity of product:

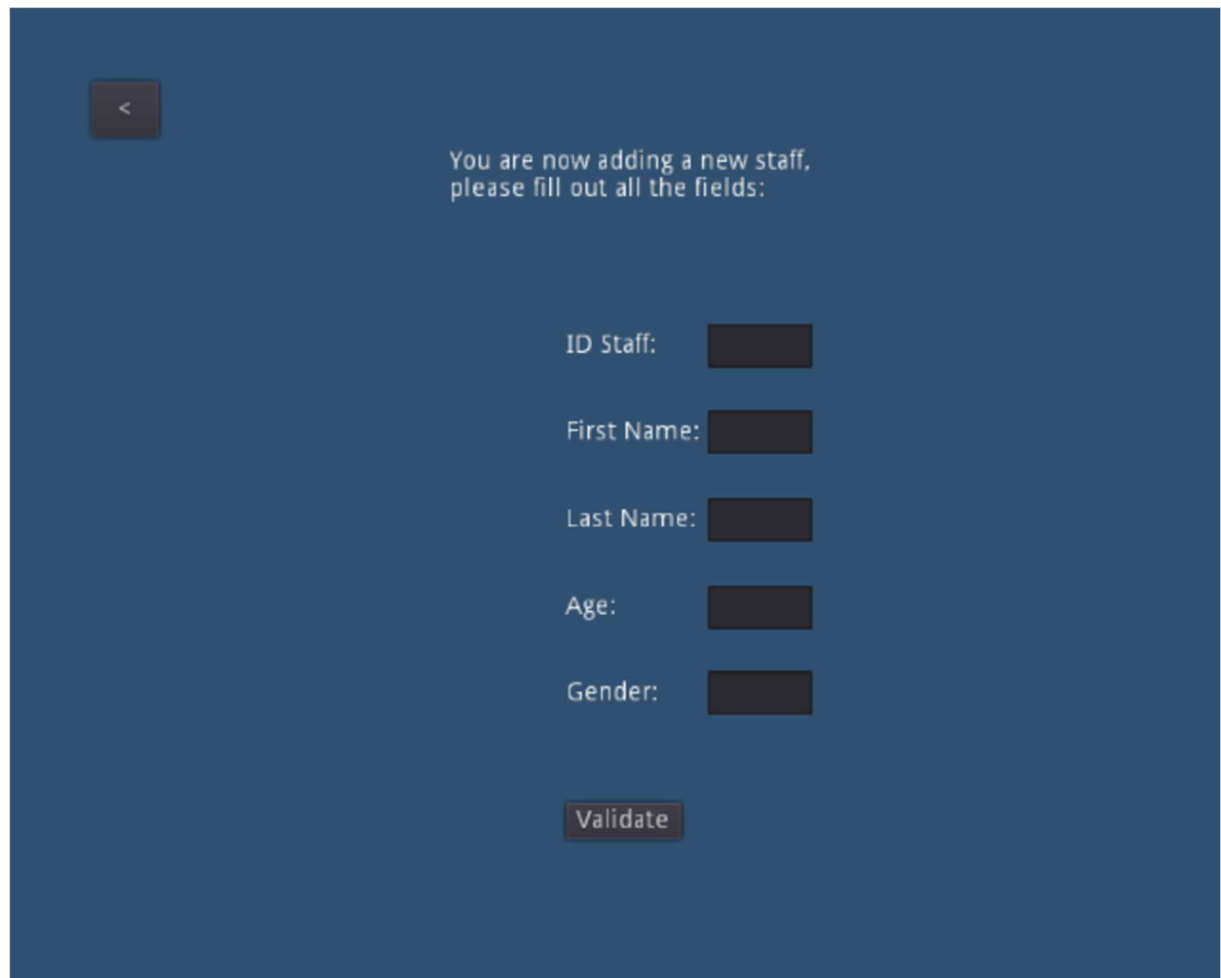
Price per unit:

Date Received:

Notes:

Validate

- For the staff, you can add new employees to the database.



<

You are now adding a new staff,
please fill out all the fields:

ID Staff:

First Name:

Last Name:

Age:

Gender:

Validate

VI. Conclusion:

In conclusion, this project has successfully addressed the issues faced by GoldenBlood, the blood center, by creating a web-based application that streamlines the blood donation process. The application has improved the efficiency, reliability, and security of the donation process, reducing errors and saving time. The implementation of data manipulation operations and queries allows for effective management and retrieval of donor, recipient, and inventory data. The logical design of tables, attribute data types, constraints, and views ensures data integrity and facilitates data manipulation. Overall, this project has made significant advancements in optimizing the blood donation workflow, ultimately benefiting both GoldenBlood and the individuals and hospitals it serves.

VII. Future work:

There are opportunities for future improvements and developments in this project. Expanding the web-based application's features is one area to focus on. This may include adding appointment reminders, checking blood type compatibility, and integrating with other systems for easier data sharing. Improving the user interface and experience would make the application more user-friendly for administrators, donors, and recipients. Collecting user feedback through testing would help identify areas for improvement. Additionally,

implementing strong data backup, recovery systems, and enhanced security measures would ensure data protection and privacy.